* What kind of language is JavaScript?

Javascript object based programming language.

It’s a interpreted language.

It’s a cross platform and light weight programming language.

Usage:

Client side validation

Dynamic behavior in UI

* What is a scripting language?

A **scripting** or **script language** is a [programming language](https://en.wikipedia.org/wiki/Programming_language) that supports **scripts:** programs written for a special [run-time environment](https://en.wikipedia.org/wiki/Run-time_environment) that [automate](https://en.wikipedia.org/wiki/Automate) the [execution](https://en.wikipedia.org/wiki/Execution_(computing)) of tasks that could alternatively be executed one-by-one by a human operator. Scripting languages are often interpreted (rather than [compiled](https://en.wikipedia.org/wiki/Compiler)).

Environments that can be automated through scripting include [software applications](https://en.wikipedia.org/wiki/Software_application), [web pages](https://en.wikipedia.org/wiki/Web_page)within a [web browser](https://en.wikipedia.org/wiki/Web_browser), the [shells](https://en.wikipedia.org/wiki/Shell_script) of [operating systems](https://en.wikipedia.org/wiki/Operating_system) (OS), [embedded systems](https://en.wikipedia.org/wiki/Embedded_system).

Semicolon not mandatory but if multiple statement are written in single line then semicolon in mandatory.

Comment

Single line comment //

Multiline comment

/\*

\* This is a multiline comment in JavaScript

\* It is very similar to comments in C Programming

\*/

If scripting is not enabled in browser following statement will be executed

<noscript>

Sorry...JavaScript is needed to go ahead.

</noscript>

We can write script in 3 ways

1. In side <head> </head> tag 🡪 generally code is written in a function which is called by event
2. Inside <body> </body> tag-🡪 written when we want code to get executed as soon as page is loaded.
3. Write in external file include in header.

<script type="text/javascript" src="filename.js" ></script>

**JavaScript data types**

* String ex: "This text string"
* Number ex: 123, 120.50
* Boolean true or false
* null
* undefined
* object - composite type

String, Number, Boolean, undefined, null are primitive data types

**Java script variable declaration**

JavaScript is a untyped language. Any type of value can be assigned to a variable.

var name = "Ali";

var money;

money = 2000.50;

Scope of variables

* local 🡪 available to the block of code
* global 🡪 available throughout the application

**Operators**

* Arithmetic Operators (**+ (Addition), - (Subtraction), \* (Multiplication), / (Division), % (Modulus), ++ (Increment), -- (Decrement)**)
* Comparison Operators (**= = (Equal), != (Not Equal), > (Greater than), < (Less than), >= (Greater than or Equal to), <= (Less than or Equal to)**)
* Logical (or Relational) Operators (**&& (Logical AND), || (Logical OR), ! (Logical NOT), & (Bitwise AND), | (BitWise OR), ^ (Bitwise XOR), ~ (Bitwise Not), << (Left Shift), >> (Right Shift), >>> (Right shift with Zero)**)

A=2,B=3

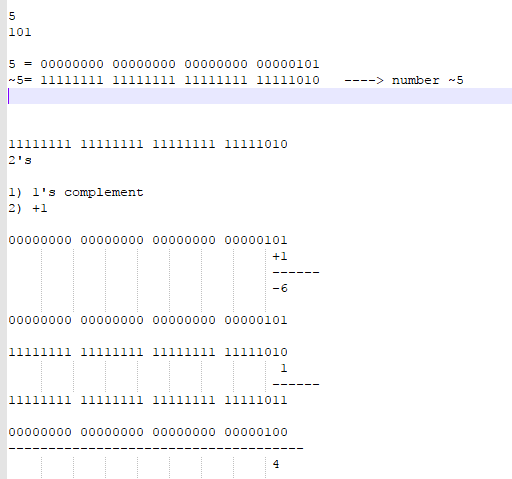
A& B = 10 A|B=10 A^B=10 ~A=10

11 11 11 ~B =11

10=2 11=3 01=1

* Assignment Operators (**= (Simple Assignment ), += (Add and Assignment), −= (Subtract and Assignment), \*= (Multiply and Assignment), /= (Divide and Assignment), %= (Modules and Assignment)**)

 Same logic applies to Bitwise operators so they will become like <<=, >>=, >>=, &=, |= and ^=.



* Conditional (or ternary) Operators (**? : (Conditional )**)
* **typeof Operator**

it’s a unary operator

|  |  |
| --- | --- |
| **Type** | **String Returned by typeof** |
| Number | "number" |
| String | "string" |
| Boolean | "boolean" |
| Object | "object" |
| Function | "function" |
| Undefined | "undefined" |
| Null | "object" |

Ex:

var a = 10;

var b = "String";

var linebreak = "<br />";

result = (typeof b == "string" ? "B is String" : "B is Numeric");

document.write("Result => ");

document.write(result);

document.write(linebreak);

result = (typeof a == "string" ? "A is String" : "A is Numeric");

document.write("Result => ");

document.write(result);

document.write(linebreak);

If else ,switch case ,for, while and do while ,break, continue same as Java

**For In**

for (variablename in object){

statement or block to execute

}

In each iteration, one property from **object** is assigned to **variablename** and this loop continues till all the properties of the object are exhausted.

**Function**

<html>

<head>

<script type="text/javascript">

function sayHello(name, age)

{

document.write (name + " is " + age + " years old.");

}

</script>

</head>

<body>

<p>Click the following button to call the function</p>

<form>

<input type="button" onclick="sayHello('Zara', 7)" value="Say Hello">

</form>

<p>Use different parameters inside the function and then try...</p>

</body>

</html>

**Nested Function**

function hypotenuse(a, b) {

function square(x) { return x\*x; }

return Math.sqrt(square(a) + square(b));

}

**Function Constructor**

Dynamically function can be created using Function ( ) constructor

<script type="text/javascript">

var variablename = new Function(Arg1, Arg2..., "Function Body");

</script>

var func = new Function("x", "y", "return x\*y;");

above function can be called as func(10,20);

**Function Literals**

The syntax for a **function literal** is much like a function statement, except that it is used as an expression rather than a statement and no function name is required.

Syntactically, you can specify a function name while creating a literal function as follows.

<script type="text/javascript">

<!--

var variablename = function FunctionName(Argument List){

Function Body

};

//-->

</script>

But this name does not have any significance, so it is not worthwhile.

What is an Event ?

JavaScript's interaction with HTML is handled through events that occur when the user or the browser manipulates a page. When the page loads, it is called an event. When the user clicks a button, that click too is an event. Other examples include events like pressing any key, closing a window, resizing a window, etc.Developers can use these events to execute JavaScript coded responses, which cause buttons to close windows, messages to be displayed to users, data to be validated, and virtually any other type of response imaginable.Events are a part of the Document Object Model (DOM) Level 3 and every HTML element contains a set of events which can trigger JavaScript Code.

**Type coersion**

Autometic convertion of type is called type coersion.Java script is forgiving language so it happens in JS.

25+true 🡪 26

25+false 🡪25

25 \* false 🡪 0

5-“1” 🡪4

**Operators which converts to number**

**\*,/,- and %**

**+ operator behaves differently .its first priority is to convert to string.**

**5+”1”🡪 51**

**Truthyness of types**

**There are only 7 values are false except everything else is true.**

* **False**
* **0**
* **-0**
* **“”**
* **Null**
* **Undefined**
* **NaN**